ELSEVIER

Annals of Pure and Applied Logic 75 (1995) 67–77

ANNALS OF PURE AND APPLIED LOGIC

Relating the bounded arithmetic and polynomial time hierarchies

Samuel R. Buss*,[1]

Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112, USA

Received 22 May 1993; revised 30 November 1994; communicated by S. Artemov

Abstract

The bounded arithmetic theory S_2 is finitely axiomatized if and only if the polynomial hierarchy provably collapses. If T_2^i equals S_2^{i+1} then T_2^i is equal to S_2 and proves that the polynomial time hierarchy collapses to Σ_{i+3}^p, and, in fact, to the Boolean hierarchy over Σ_{i+2}^p and to $\Sigma_{i+1}^p/poly$.

1. Introduction

Theories of bounded arithmetic are theories of arithmetic obtained by putting restrictions on induction axioms; namely, allowing induction only for certain classes, Σ_i^b, of bounded formulas, and using polynomial, or length, induction (PIND or LIND) in place of successor induction (IND). The most important subtheories of bounded arithmetic are the theories S_2^i, axiomatized with Σ_i^b-PIND (or equivalently, Σ_i^b-LIND, if $i \geqslant 1$), and the theories T_2^i, axiomatized by Σ_i^b-IND. The following inclusions are known for these theories:

$$S_2^0 \subsetneq T_2^0 \subseteq S_2^1 \subseteq T_2^1 \subseteq S_2^2 \subseteq T_2^2 \subseteq \cdots$$

and their union is the theory $S_2 = T_2$ [2]. However, with the exception of $S_2^0 \neq T_2^0$ (see [13]), it is not known whether the rest of the theories of bounded arithmetic are distinct. It is a well-known fact that S_2^i and T_2^i are finitely axiomatized for $i > 0$, and thus it is immediate that this hierarchy of theories collapses if and only if S_2 is finitely axiomatized. This latter condition is equivalent to $I\Delta_0 + \Omega_1$ being finitely axiomatized (see [11, 14] for this alternate, and original, approach to bounded arithmetic).

* Email: sbuss@ucsd.edu.

[1] Supported in part by NSF grant DMS-9205181.

SSDI 0168-0072(94)00057-3

There are close connections between theories of bounded arithmetic and the polynomial hierarchy. First, the class of predicates definable by Σ_i^b (or Π_i^b) formulas is precisely the class of predicates in the ith level Σ_i^p (or Π_i^p, respectively) of the polynomial hierarchy. For instance, S_2^1 and T_2^1 are axiomatized with their induction axioms restricted to NP-predicates (since NP $= \Sigma_1^p$ is the class of predicates definable by Σ_1^b-formulas). Second, it is known that the Σ_i^b-definable functions of S_1^i are precisely the $\square_i^p$-functions, which are the functions which are polynomial time computable with an oracle for Σ_{i-1}^b. For instance, the Σ_1^b-definable functions of S_2^1 are precisely the polynomial time computable functions.

Since it is open whether the polynomial time hierarchy collapses, it is natural to ask whether there is any connection between the possible collapses of the hierarchy of bounded arithmetic theories and the polynomial hierarchy. This question has already been partially answered by the work of Krajíček et al. [10] who showed that if $T_2^i = S_2^{i+1}$ for any $i \geqslant 1$, then the polynomial hierarchy collapses with $\Sigma_{i+2}^p = \Pi_{i+2}^p$ (in fact, they show that in this case, $\Sigma_{i+1}^p \subset \Delta_i^p/poly$).

The main results of this paper strengthen the results of Krajíček et al. by proving that if $T_2^i = S_2^{i+1}$ holds, then the following conditions must hold: (1) $T_2^i = S_2$, so that the hierarchy of bounded arithmetic theories collapses, and (2) T_2^i can prove that the polynomial time hierarchy collapses to $\mathscr{B}(\Sigma_{i+2}^p)$ and to $\Sigma_{i+1}^p/poly$, where $\mathscr{B}(\Sigma_{i+2}^p)$ is the class of Boolean combinations of Σ_{i+2}^b-predicates. Our proofs are easier, in a combinatorial sense, than the proofs of [10] and this makes it possible to formalize them in T_2^i.

We believe that the results of this paper are nearly the strongest that are obtainable relating the possibility that $T_2^i = S_2^{i+1}$ to the possible collapse of the polynomial time hierarchy – at least with current techniques. To support this belief, consider the three conditions:

(α) the polynomial hierarchy collapses,

(β) S_2 proves that the polynomial hierarchy collapses,

(γ) S_2 is finitely axiomatized.

Our results show that (β) and (γ) are equivalent; however, we do not expect to show that (α) is equivalent to (γ) using current techniques. The reason for this is that (α) is a Σ_2^0-condition whereas, since (β) is a Σ_1^0-condition, the results of the current paper show that (γ) is a Σ_1^0-condition; and, based on the history of attempts to solve the P versus NP problem, it seems to be difficult even to establish that the collapse of the polynomial time hierarchy is equivalent to a natural Σ_1^0-condition like (γ).

It is known that S_2^{i+1} is conservative over T_2^i with respect to $\forall\Sigma_{i+1}^b$-sentences [3]. On the other hand, the axioms of S_2^{i+1} can be expressed as $\forall\Pi_{i+2}^b$-sentences (in this formulation, an induction axiom of S_2^{i+1} will become a $\forall\Pi_{i+2}^b$-formula with a sharply bounded existential quantifier in its outermost block of bounded universal quantifiers). Thus saying S_2^{i+1} is Π_{i+2}^b-conservative over T_2^i is equivalent to saying that $S_2^{i+1} = T_2^i$.

An open problem is to try to relate the condition $S_2^i = T_2^i$ to the possible collapse of the polynomial hierarchy. Krajíček [9] shows that if $S_2^i = T_2^i$, then the set $\leqslant_m^p(\Sigma_i^p)$ of

predicates logspace, Turing reducible to Σ_i^p is equal to the set $\leqslant_T^p(\Sigma_i^p)$ of predicates polynomial time, Turing reducible to Σ_i^p. However, it is open whether this last condition implies the polynomial hierarchy collapses. See [4–6] for more on this connection.

The prerequisites for reading this paper are a basic knowledge of bounded arithmetic theories as contained in [2]. The reader would also benefit from knowledge of [10, 3]. In the next section we will review the necessary background material needed from [10].

After preparing the first draft of this paper, we learned that D. Zambella has independently discovered the main results of this paper [15].

2. The KPT witnessing theorem for T_2^i

There are two important witnessing theorems for T_2^i. The first follows from the 'Main Theorem' for S_2^{i+1} and the fact that S_2^{i+1} is Σ_{i+1}^b-conservative over T_2^i: this witnessing theorem states that the Σ_{i+1}^b-definable functions of T_2^i are precisely the functions which can be computed in polynomial time with a Σ_i^b-oracle (i.e., the $\Box_{i+1}^p$-functions). The second witnessing theorem puts a necessary condition on the Σ_{i+2}^b- and Σ_{i+3}^p-definable functions of T_2^i; we call this the 'KPT witnessing theorem'. It is this latter witnessing theorem that we need for our proofs:

Theorem 1. *Let $i \geqslant 1$. Suppose $T_2^i \vdash (\forall x)(\exists y)(\forall z)B(x,y,z)$, where B is a $\exists \Pi_i^b$-formula, with only x, y, z as free variables. There exists $k > 0$ and functions $f_1, \ldots, f_k$ such that each f_m is m-ary and is Σ_{i+1}^b-definable by T_2^i and such that*

$$T_2^i \vdash B(a, f_1(a), b_1) \vee B(a, f_2(a, b_1), b_2) \vee B(a, f_3(a, b_1, b_2), b_3) \vee \cdots$$
$$\vee B(a, f_k(a, b_1, b_2, \ldots, b_{k-1}), b_k).$$

For $i = 0$, the same result holds for PV_1 in place of T_2^0. As usual, PV_1 denotes the conservative extension of PV to first-order logic, or equivalently, PV_1 is S_2^0 or T_2^0 enlarged to have function symbols and their defining equations for all polynomial time functions.

Note that since the functions f_m are Σ_{i+1}^b-definable by T_2^i, they must be $\Box_{i+1}^p$-functions.

Theorem 1 is due to [10]; some later, related results can be found in [8, 12, 1]. We do not include a proof here.

We next use Theorem 1 to establish a consequence of the condition $T_2^i = S_2^{i+1}$. We assume that $i \geqslant 0$ and work with the theory T_2^i; when $i = 0$ our results are intended to hold for PV_1 in place of T_2^0.

Definition. A *quantified Boolean formula* is a formula constructed from Boolean connectives (say, $\wedge$, $\vee$ and $\neg$) and quantifiers ranging over Boolean values. A quantifier $(\forall p)$ or $(\exists p)$ indicates quantification allowing p to range over the values *True* and *False*.

Given a truth assignment to the free variables of a quantified Boolean formula, it is obvious how the truth value of the formula should be defined. A quantified Boolean formula is *satisfiable* if there is some truth assignment to its free variables which gives it value *True*. A Π_i^B-formula is a quantified Boolean formula which is in prenex form with i blocks of like quantifiers starting with a universal block. It is well-known that the set of satisfiable Π_i^B-formulas is Σ_{i+1}^p-complete.

Definition. Let $i \geqslant 0$. TRU^i and SAT^i are bounded arithmetic formulas which express:

$$TRU^i(\varphi, w) \Leftrightarrow \varphi \text{ codes a } \Pi_i^B\text{-formula and } w \text{ codes a satisfying assignment of } \varphi,$$

$$SAT^i(\varphi) \Leftrightarrow (\exists w \leqslant \varphi)\, TRU^i(\varphi, w).$$

In the definition of TRU^i and SAT^i we presume that quantified Boolean formulas and truth assignments are coded in some natural and efficient way by integers; we use Greek letters $\varphi, \ldots$ as variables that range over integers which are intended to code quantified Boolean formulas. Since the code of a truth assignment can w.l.o.g. always be less than the code of a formula, $SAT^i(\varphi)$ expresses the condition that φ is satisfiable. Standard bootstrapping techniques allow TRU^i to be a Δ_{i+1}^b-formula with respect to the theory T_2^i; in fact, for $i \geqslant 1$, TRU^i is a Π_i^b-formula. Hence SAT^i is a Σ_{i+1}^b-formula. Also, T_2^i can prove basic properties of the TRU^i and SAT^i predicates. Most importantly, T_2^i can prove that SAT^i is many-one complete for Σ_{i+1}^b-formulas; i.e., for any Σ_{i+1}^b-formula $A(\vec{b})$, there is a polynomial time function f so that $A(\vec{b})$ is T_2^i-provably equivalent to $SAT^i(f(\vec{b}))$.

As an application of Theorem 1, consider the formula

$$\forall\langle\varphi_0, \ldots, \varphi_n\rangle(\exists\, l \leqslant n)(\exists\langle w_0, \ldots, w_l\rangle)$$
$$[(\forall j \leqslant l) TRU^i(\varphi_j, w_j) \wedge (l < n \rightarrow \neg(\exists w_{l+1}) TRU^i(\varphi_{l+1}, w_{l+1}))]. \tag{1}$$

The meaning of formula (1) requires some explanation. First, a notation like $(\forall\langle\varphi_0, \ldots, \varphi_n\rangle)B(\vec{\varphi}, n)$ means the same as 'there is an integer φ^* which codes a sequence of Π_i^B-formulas $\varphi_0, \ldots, \varphi_n$ so that $B(\vec{\varphi}, n)$ holds'. The quantifier $(\exists l \leqslant n)$ is a sharply bounded quantifier since l can be bounded by the length of the code for $\langle\vec{\varphi}\rangle$, and the quantifiers $(\exists\langle\vec{w}\rangle)$ and $(\exists w_{l+1})$ are bounded quantifiers since each w_j may be bounded by φ_j. By using prenex operations and using the fact that l can be computed from $\langle w_0, \ldots, w_l\rangle$, formula (1) is equivalent to the formula

$$(\forall\langle\varphi_0, \ldots, \varphi_n\rangle)(\exists\langle w_0, \ldots, w_l\rangle)(\forall w_{l+1})$$
$$[(\forall j \leqslant l) TRU^i(\varphi_j, w_j) \wedge (l < n \rightarrow \neg\, TRU^i(\varphi_{l+1}, w_{l+1}))]. \tag{2}$$

which is a $\forall\exists \leqslant \forall \leqslant \Delta_{i+1}^b$-formula.

The intuitive meaning of formula (1) or (2) is, of course, that every sequence $\varphi_0, \ldots, \varphi_n$ of Π_i^B-formulas has an initial sequence of maximal length l of satisfiable

formulas. Furthermore, the formula (1) is a theorem of S_2^{i+1}. This is because S_2^{i+1} can use length induction on the Σ_{i+1}^b-formula $S(\langle\vec{\varphi}\rangle, l)$ expressing the condition that the first l formulas of the sequence are satisfiable. (An equivalent way to see this is to note that S_2^{i+1} can prove the Σ_{i+1}^b-length-maximization principle.)

Now suppose T_2^i is equal to S_2^{i+1}; in particular, T_2^i proves the formula (2). By Theorem 1, this means that there is an integer $k \geqslant 0$ and there are Σ_{i+1}^b-defined functions $f_0, \ldots, f_k$ so that, letting $\mathscr{A}(\langle\vec{\varphi}\rangle, \langle\vec{w}\rangle, w_{l+1})$ be the subformula of (2) enclosed in square brackets, we have that

$$T_2^i \vdash (\forall\langle\vec{\varphi}\rangle)[A(\langle\vec{\varphi}\rangle, f_0(\langle\vec{\varphi}\rangle), b_0) \vee A(\langle\vec{\varphi}\rangle, f_1(\langle\vec{\varphi}\rangle, b_0), b_1) \vee \cdots$$
$$\vee A(\langle\vec{\varphi}\rangle, f_k(\langle\vec{\varphi}\rangle, b_0, b_1, \ldots, b_{k-1}), b_k)] \tag{3}$$

We henceforth shall use (3) restricted to the case where $n = k$, so that the sequence $\vec{\varphi}$ is $\varphi_0, \ldots, \varphi_k$.

Without loss of generality, each f_j satisfies the following property (provably in T_2^i): whenever $TRU^i(\varphi_r, b_r)$ holds for $r = 0, \ldots, j-1$, then the value $f_j(\langle\vec{\varphi}\rangle, b_0, \ldots, b_{j-1})$ is the Gödel number of a sequence $\langle v_0, \ldots, v_{l-1}\rangle$ of length $l \geqslant j$ so that $TRU^i(\varphi_r, v_r)$ holds for all $r = 0, \ldots, l-1$.

Recall that β represents the Gödel β function so that $\beta(i, w)$ is equal to the ith integer in the sequence coded by w. Define

$$g_j(\varphi_0, \ldots, \varphi_k, w_0, \ldots, w_{j-1}) = \beta(j+1, f_j(\langle\varphi_0, \ldots, \varphi_k\rangle, w_0, \ldots, w_{j-1})).$$

Suppose that $\varphi_0, \ldots, \varphi_k$ are codes for satisfiable Π_i^B-Boolean formulas and let $w_0, \ldots, w_k$ be satisfying assignments. Define $b_0, b_1, \ldots$ inductively as follows: if $f_j(\langle\vec{\varphi}\rangle, b_0, \ldots, b_{j-1})$ is a sequence of length $l+1 \leqslant k$, then let b_j equal w_{l+1}. It is obvious that whenever $f_j(\langle\vec{\varphi}\rangle, b_0, \ldots, b_{j-1})$ has length $l+1 \leqslant k$ then b_j gives a "counterexample" so that $A(\langle\vec{\varphi}\rangle, f_j(\langle\vec{\varphi}\rangle, b_0, \ldots, b_{j-1}), b_j)$ is false. Now, by (2), there is some $j \leqslant k$ for which $f_j(\langle\vec{\varphi}\rangle, b_0, \ldots, b_{j-1})$ has length $k+1$. Let j_0 be the least value such that $f_{j_0}(\langle\vec{\varphi}\rangle, b_0, \ldots, b_{j-1})$ has length $\geqslant j_0 + 1$. It must be that $TRU^i(\varphi_{j_0}, g_{j_0}(\vec{\varphi}, w_0, \ldots, w_{j_0-1}))$ holds. This argument formalizes in T_2^i and thus we have proven:

Lemma 2. *Suppose $T_2^i = S_2^{i+1}$. Then there is $k \geqslant 0$ and there are Σ_{i+1}^b-definable functions $g_0, \ldots, g_k$ of T_2^i so that*

$$T_2^i \vdash (\forall\varphi_0, \ldots, \varphi_k)(\forall w_0, \ldots, w_k)\Big[\bigwedge_{j=0}^{k} TRU^i(\varphi_j, w_j)$$
$$\rightarrow TRU^i(\varphi_0, g_0(\vec{\varphi}))$$
$$\vee TRU^i(\varphi_1, g_1(\vec{\varphi}, w_0))$$
$$\vee TRU^i(\varphi_2, g_2(\vec{\varphi}, w_0, w_1))$$
$$\vee \cdots \vee TRU^i(\varphi_k, g_k(\vec{\varphi}, w_0, \ldots, w_{k-1}))\Big]$$

3. Collapsing bounded arithmetic

In this and the next section, we examine consequences of the condition $T_2^i = S_2^{i+1}$. In this section we show that this implies that S_2 collapses to T_2^i.

Our point of departure is Lemma 2; we henceforth fix k and $g_0, \ldots, g_k$. This lemma states that at least one of the functions g_j can find a satisfying assignment for φ_j using only the vector $\vec{\varphi}$ and arbitrary satisfying assignments $w_0, \ldots, w_{j-1}$. However, it need not always be the same g_j that succeeds in this way; different vectors of formulas $\vec{\varphi}$ and even different witnesses $\vec{w}$ may cause different g_j's to succeed. We define *Succeed By*$(l, \vec{\varphi}, \vec{w})$ to be the following formula which states that one of the first $l+1$ g's succeeds in this way; namely, it is defined as

$$SucceedBy(l, \vec{\varphi}, \vec{w}) \Leftrightarrow \bigvee_{j=0}^{k} [j \leqslant l \wedge TRU^i(\varphi_j, g_j(\vec{\varphi}, w_0, \ldots, w_{j-1}))].$$

Our first goal is to show that $\Sigma_{i+1}^p = \Pi_{i+1}^p/poly$ where the '*poly*' means that polynomial amount of advice is needed. As a preliminary to defining what constitutes advice, we define 'preadvice' by letting $PreAdvice^i(a, \langle \varphi_{l+1}, \ldots, \varphi_k \rangle)$ be the formula

$$\bigwedge_{j=0}^{k} (l < j \rightarrow \varphi_j < 2^{|a|}) \wedge (\forall \langle \varphi_0, \ldots, \varphi_l \rangle)(\forall \langle w_0, \ldots, w_l \rangle)$$
$$[VTRU^i(\langle \vec{\varphi} \rangle, \langle \vec{w} \rangle, a) \rightarrow SucceedBy(l, \vec{\varphi}, \vec{w})],$$

where $VTRU^i(\langle \varphi_0, \ldots, \varphi_k \rangle, \langle w_0, \ldots, w_l \rangle, a)$ abbreviates

$$(\forall j \leqslant l)(TRU^i(\varphi_j, w_j) \wedge w_j \leqslant \varphi_j \wedge \varphi_j < 2^{|a|}).$$

Several points to note are: firstly, in defining *PreAdvice* we are continuing our practice of letting variables φ_j represent integers that must code Π_i^B formulas; secondly, the value of l is determined by the second argument to *PreAdvice* (k is fixed and l varies, namely, l equals $k+1$ minus the length of the sequence coded by the second argument of $PreAdvice^i$); thirdly, the quantifiers are bounded quantifiers since the φ_j's and w_j's are bounded by $2^{|a|}$. The reason for bounding everything by $2^{|a|}$ is that we need only define 'advice' that works for φ's with $|\varphi| \leqslant a$ for a an arbitrary integer. Also note that $Preadvice^i$ is a Π_{i+1}^b-formula.

We can now define 'advice' for formulas of length $\leqslant |a|$ by

$$Advice^i(a, \langle \varphi_{l+1}, \ldots, \varphi_k \rangle)$$
$$\Leftrightarrow PreAdvice^i(a, \langle \varphi_{l+1}, \ldots, \varphi_k \rangle) \wedge \neg (\exists \varphi_l) PreAdvice^i(a, \langle \varphi_l, \ldots, \varphi_k \rangle).$$

Note that φ_l is bounded by $2^{|a|}$; thus $Advice^i$ is a Π_{i+2}^b formula. The next lemma shows that T_2^i can prove that there always does exist advice:

Lemma 3. *Suppose* $T_2^i = S_2^{i+1}$. *Then*

$$T_2^i \vdash (\forall a)(\exists \langle \vec{\varphi} \rangle) Advice^i(a, \langle \vec{\varphi} \rangle).$$

Proof. First, note that Lemma 2 implies that T_2^i proves that $PreAdvice^i(a, \langle\ \rangle)$ holds. Since k is a constant, it follows (without using induction) that there is a least l such that $(\exists \langle \varphi_{l+1}, \ldots, \varphi_k \rangle) PreAdvice^i(a, \langle \vec{\varphi} \rangle)$ holds. For this l, any 'preadvice' is actually advice. □

Next we give the key lemma that shows how 'advice' can be used to make Σ^b_{i+1}-IND hold and the polynomial time hierarchy collapse, probably in T_2^i.

Lemma 4. *Suppose $T_2^i = S_2^{i+1}$. Then T_2^i proves*

$$\begin{aligned} &Advice^i(a, \langle \varphi_{l+1}, \ldots, \varphi_k \rangle) \wedge \varphi_l < 2^{|a|} \\ &\quad \rightarrow [\neg SAT^i(\varphi_l) \leftrightarrow (\exists \langle \varphi_0, \ldots, \varphi_{l-1} \rangle)(\exists \langle w_0, \ldots, w_{l-1} \rangle) \\ &\qquad\qquad \{ VTRU^i(\langle \vec{\varphi} \rangle, \langle w_0, \ldots, w_{l-1} \rangle, a) \\ &\qquad\qquad \wedge \neg SucceedBy(l-1, \vec{\varphi}, \vec{w}) \\ &\qquad\qquad \wedge \neg TRU^i(\varphi_l, g_l(\varphi_0, \ldots, \varphi_k, w_0, \ldots, w_{l-1})) \}]. \end{aligned}$$

Proof. Let $RHS(\varphi_l, \langle \varphi_{l+1}, \ldots, \varphi_k \rangle, a))$ denote the formula on the right-hand side of the $\leftrightarrow$ connective in the formula above; we often suppress the variables $\varphi_{l+1}, \ldots, \varphi_k$ and a that occur freely in RHS and write just $RHS(\varphi_l)$.

We shall argue informally in T_2^i to prove the lemma. Suppose $\varphi_l, \ldots, \varphi_k \leqslant 2^{|a|}$ are formulas and that $Advice^i(a, \langle \varphi_{l+1}, \ldots, \varphi_k \rangle)$ holds. The latter condition obviously implies that $\neg PreAdvice^i(a, \langle \varphi_l, \ldots, \varphi_k \rangle)$. By the definition of $PreAdvice$, there must exist Π_i^B-formulas $\varphi_0, \ldots, \varphi_{l-1}$ satisfied by witnesses $w_0, \ldots, w_{l-1}$ such that $SucceedBy(l-1, \vec{\varphi}, \vec{w})$ is forced to be false. First suppose that φ_l is not satisfiable. Then clearly $TRU^i(\varphi_l, g_l(\vec{\varphi}, \vec{w}))$ must be false. Thus $RHS(\varphi_l)$ follows from $\neg SAT^i(\varphi_l)$. Second, suppose that φ_l is satisfiable. By $PreAdvice^i(a, \langle \varphi_{l+1}, \ldots, \varphi_k \rangle)$, it must be that $SucceedBy(l, \vec{\varphi}, \vec{w})$ holds. On the other hand, $SucceedBy(l-1, \vec{\varphi}, \vec{w})$ is false. Thus $TRU^i(\varphi_l, g_l(\vec{\varphi}, \vec{w}))$ is forced to be true and we have shown that $SAT^i(\varphi_l)$ implies $\neg RHS(\varphi_l)$. □

In the subformula RHS, the leading existential quantifiers are actually bounded existential quantifiers since the formulas φ_j and their witnesses w_j are bounded by $2^{|a|}$. This means that $RHS(\varphi_l)$ is a Σ^b_{i+1}-formula.

Lemma 5. *Suppose $T_2^i = S_2^{i+1}$. Then $T_2^i \vdash \Sigma^b_{i+1}$-IND and $T_2^i = T_2^{i+1}$.*

Proof. The proof is based on the fact that $SAT^i(\cdots)$ is complete for Σ^b_{i+1}-formulas and is also equivalent on bounded ranges to the Π^b_{i+1}-formula $\neg RHS(\cdots)$ (under the assumption that $T_2^i = S_2^{i+1}$, as always). Indeed, for any Σ^b_{i+1}-formula $B(c, \vec{d})$, there is a polynomial time and Σ_1^b-computable function $f(c, \vec{d})$ so that $B(c, \vec{d})$ is T_2^i-provably equivalent to $SAT^i(f(c, \vec{d}))$. The induction axiom for the formula $B(c, \vec{d})$ can be

expressed as

$$B(0,\vec{d}) \wedge (\forall x)(B(x,\vec{d}) \to B(x+1,\vec{d})) \to B(c,\vec{d}).$$

Let us prove this by reasoning informally in T_2^i which is presumed to equal S_2^{i+1}. Considering particular values for c and $\vec{d}$, there is a value a so that $f(x,\vec{d}) < 2^{|a|}$ for all $x \leqslant c$. Let $\varphi_{l+1},\ldots,\varphi_k$ be formulas such that $Advice^i(a,\langle\varphi_{l+1},\ldots,\varphi_k\rangle)$ holds. Then, with these parameters, by Lemma 4, we have that the Σ^b_{i+1}-formula $B(x,\vec{d})$ is equivalent to the Π^b_{i+1}-formula $\neg RHS(f(x,\vec{d}))$ for all $x \leqslant c$. Now, it is known that S_2^{i+1} proves Δ^b_{i+1}-IND and the usual proof (see [2, Theorem 2.22]) shows that $T_2^i = S_2^{i+1}$ proves induction for B, since B is "Δ^b_{i+1} with parameters" on the range $0 \leqslant x \leqslant c$. □

Iterating the method of this proof, we obtain:

Theorem 6. *If $T_2^i = S_2^{i+1}$, then $T_2^i = S_2$. Thus, if $T_2^i = S_2^{i+1}$, then S_2 is finitely axiomatized.*

Also, if $PV_1 = S_2^1(PV)$, then $PV_1 = S_2(PV)$.

Proof. Analogous to the method of proof of Lemma 5, we must show that any bounded formula is equivalent to a Σ^b_{i+1}-formula with parameters, where the parameters vary with the range of the induction variable. From this, using Lemma 5, it will follow that T_2^i proves induction for any bounded formula.

We do the case of $B(c,\vec{d}) \in \Sigma^b_{i+2}$ in some detail. We may suppose that $B(x,\vec{d})$ is of the form $(\exists y \leqslant t(x,d))C(x,y,\vec{d})$ for some Π^b_{i+1}-formula C. We argue informally in T_2^i. By the method of Lemma 5, there is an a, given by a polynomial time function $a = a(c,\vec{d})$ of c and $\vec{d}$, and there is a polynomial time function f, so that for all $x \leqslant c$, and $y \leqslant t(x,\vec{d})$ and for advice $\langle\vec{\varphi}\rangle$ satisfying $Advice^i$, the Π^b_{i+1}-formula $C(x,y,\vec{d})$ is equivalent to the Σ^b_{i+1}-formula $RHS(f(x,y,\vec{d}),\langle\vec{\varphi}\rangle, a(c,\vec{d}))$. Thus, for $0 \leqslant x \leqslant c$, $B(x,\vec{d})$ is equivalent to a Σ^b_{i+1}-formula, and full induction holds for B up to c by Lemma 5. Hence $T_2^i = T_2^{i+2}$.

A slight modification of the construction of the last paragraph shows that if $A(\vec{x})$ is a Σ^b_{i+2}-formula (respectively, a Π^b_{i+2}-formula, then there is a polynomial growth rate function $a(c)$ and a Σ^b_{i+1}-formula (respectively, Π^b_{i+1}-formula) $A^*(\vec{x},\varphi^*,a(c))$ such that for all $\vec{x}$ such that $\max\{\vec{x}\} \leqslant c$ and all $\langle\vec{\varphi}\rangle$ such that $Advice^i(a(c),\langle\vec{\varphi}\rangle)$, $A(\vec{x})$ is equivalent to $A^*(\vec{x},\langle\vec{\varphi}\rangle,a(c))$, provably in T_2^i. This further implies that if $A(\vec{x})$ is a Σ^b_{i+3}-formula, then A^* may be taken to be a Σ_{i+2}-formula, because, if $A(\vec{x})$ is $(\exists y \leqslant t(\vec{x}))B(\vec{x},y)$, then there is a Σ^b_{i+2}-formula B^* so that $A(\vec{x})$ will be equivalent to $(\exists y \leqslant t(\vec{x}))B^*(\vec{x},\langle\vec{\varphi}\rangle,a)$ for a given by a polynomial growth rate function of $c \geqslant \max\vec{x}$ and for $\langle\vec{\varphi}\rangle$ such that $Advice^i(a,\langle\vec{\varphi}\rangle)$. This fact is sufficient to imply that $T_2^i = T_2^{i+3}$.

By iterating the above method of proof, one can show that T_2^i is equal to all of S_2. We shall leave the details of this to the reader, and remark instead that an alternative proof is given by Theorem 7 below where it is shown that T_2^i can prove that every bounded formula is equivalent to a Boolean combination of Σ^b_{i+2}-formulas without

any additional parameters or advice. Then since $T_2^i = T_2^{i+2} = S_2^{i+2}$ and S_2^{i+2} proves induction for Boolean combinations of Σ_{i+2}^b-formulas [3], it follows that $T_2^i = S_2$. □

4. Collapsing the polynomial hierarchy

All the work of this section is predicated on the condition that $T_2^i = S_2^{i+1}$. We have shown above that if $T_2^i = S_2^{i+1}$, then T_2^i proves that the Σ_{i+2}^p-predicates are contained in $\Sigma_{i+1}^p/poly$. From this, the methods of Karp–Lipton [7] imply that the entire polynomial time hierarchy is contained in $\Sigma_{i+1}^p/poly$ and in $\Pi_{i+1}^p/poly$; furthermore, the proof of this containment can be formalized in T_2^i. The methods of Krap–Lipton also imply immediately that the polynomial hierarchy collapses to $\Sigma_{i+3}^p = \Pi_{i+3}^p$. However, we shall prove an somewhat stronger result; namely, if $T_2^i = S_2^{i+1}$, then every polynomial hierarchy predicate (i.e., bounded formula) is T_2^i-provably equivalent to a Boolean combination of Σ_{i+2}^b-formulas.

To prove this, it will suffice to prove that every Σ_{i+3}^b-formula is equivalent to a Boolean combination of Σ_{i+2}^b-formulas. Let $A(b)$ be an arbitrary Σ_{i+3}^b-formula. From the previous section, we know that T_2^i proves that $A(b)$ is equivalent to

$$(\exists \langle \vec{\varphi} \rangle)[Advice^i(a(b), \langle \vec{\varphi} \rangle) \wedge A^*(b, \langle \vec{\varphi} \rangle)] \tag{4}$$

and to

$$(\forall \langle \vec{\varphi} \rangle)[Advice^i(a(b), \langle \vec{\varphi} \rangle) \rightarrow A^*(b, \langle \vec{\varphi} \rangle)], \tag{5}$$

where A^* is a Σ_{i+2}^b-formula and $a = a(b)$ is function of sufficiently large polynomial growth rate. Unfortunately, $Advice^i$ is a Π_{i+2}^b-formula and the quantifier complexity of these equivalent formulations of $A(b)$ is higher than we desire; namely, formula (4) is a Σ_{i+3}^b-formula and formula (5) is a Π_{i+3}^b-formula. This implies that every bounded formula is Δ_{i+3}^b with respect to T_2^i, but we wish to prove a yet stronger result.

To reduce the complexity of these formulas we would like to use $PreAdvice^i$ in place of $Advice^i$. However, this can not be done directly since if $\langle \vec{\varphi} \rangle$ satisfies $PreAdvice^i$, then it is not necessarily true that $A^*(b, \langle \vec{\varphi} \rangle)$ is equivalent to $A(b)$. Instead, we look for a longest vector $\langle \vec{\varphi} \rangle$ which satisfies $PreAdvice^i$; namely, consider the formula $A'(b)$ defined as:

$$\begin{aligned}
&(\exists \langle \varphi_1, \ldots, \varphi_k \rangle)[PreAdvice^i(a(b), \langle \varphi_1, \ldots, \varphi_k \rangle) \wedge A^*(b, \langle \varphi_1, \ldots, \varphi_k \rangle)] \\
&\quad \vee \bigvee_{l=2}^{k} \{\neg(\exists \langle \varphi_{l-1}, \ldots, \varphi_k \rangle) PreAdvice^i(a(b), \langle \varphi_{l-1}, \ldots, \varphi_k \rangle) \\
&\quad \wedge (\exists \langle \varphi_l, \ldots, \varphi_k \rangle)[PreAdvice^i(a(b), \langle \varphi_l, \ldots, \varphi_k \rangle) \wedge A^*(b, \langle \varphi_l, \ldots, \varphi_k \rangle)]\}.
\end{aligned}$$

We claim that $A'(b)$ is equivalent to $A(b)$. The proof of this now quite easy. First, there must exist a least $l \geqslant 1$ such that there exists $\langle \varphi_l, \ldots, \varphi_k \rangle$ which satisfies $PreAdvice^i$. Second, if $PreAdvice^i(\langle \varphi_l, \ldots, \varphi_k \rangle)$ holds and if there is no $\langle \varphi'_{l-1}, \ldots, \varphi'_k \rangle$ which

satisfies $PreAdvice^i$, then clearly $\langle \varphi_1, \ldots, \varphi_k \rangle$ satisfies $Advice^i$. And for this advice, $A^*(b, \langle \vec{\varphi} \rangle)$ is equivalent to $A(b)$.

Since $PreAdvice^i$ is a Π^b_{i+1}-formula and A^* is a Σ^b_{i+2}-formula, A' is a Boolean combination of Σ^b_{i+2}-formulas. This establishes:

Theorem 7. *If $T^i_2 = S^{i+1}_2$, then every bounded formula is T^i_2-provably equivalent to a Boolean combination of Σ^b_{i+2} formulas. In other words, if $T^i_2 = S^{i+1}_2$, then the polynomial hierarchy T^i_2-provably collapses to (a finite level of) the Boolean hierarchy over Σ^b_{i+2}. Also, in this case, T^i_2 proves that the polynomial time hierarchy collapses to $\Sigma^p_{i+1}/poly$.*

If $PV_1 = S^1_2(PV)$, then every bounded formula is PV_1-provably equivalent to a Boolean combination of Σ^b_2-formulas, so the polynomial time hierarchy provably collapses to the Boolean hierarchy over Σ^p_2. Also, in this case, PV_1 proves that the polynomial time hierarchy collapses to NP/poly.

It should be noted again that [10] have shown that if $T^{i+1}_2 = S^{i+2}_2$ then the polynomial hierarchy collapses to $\Sigma^p_{i+2} = \Pi^p_{i+2}$ and to $\Delta^p_{i+1}/poly$: we do not know how to prove that this stronger collapse would be T^i_2-provable.

References

[1] S.R. Buss, The witness function method and fragments of Peano arithmetic, in: D. Prawitz et al., eds., Proc. Ninth Internat. Congress on Logic, Methodology and Philosophy of Science (North-Holland, Amsterdam, 1994) 29–68.

[2] S.R. Buss, Bounded Arithmetic, Bibliopolis, 1986. Revision of 1985 Princeton University Ph.D. Thesis.

[3] S.R. Buss, Axiomatizations and conservation results for fragments of bounded arithmetic, in: Logic and Computation, Proc. of a Workshop held at Carnegie–Mellon University, 1987, Vol. 106 of Contemporary Mathematics (American Mathematical Society, Providence, RI, 1990) 57–84.

[4] S.R. Buss and L. Hay, On truth-table reducibility to SAT, Inform. Comput. 91 (1991) 86–102.

[5] S.R. Buss and J. Krajíček, An application of Boolean complexity to separation problems in bounded arithmetic, Proc. London Math. Soc. 69 (1994) 1–21.

[6] R. Chang and J. Kadin, The boolean hierarchy and the polynomial hierarchy: a closer connection, in: Proc. Fifth Annual Structure in Complexity Conference (IEEE Computer Society Press, Silver Spring, MD, 1990) 169–178.

[7] R.M. Karp and R.J. Lipton, Turing machines that take advice, L'Enseignement Mathematique 28 (1982) 191–209.

[8] J. Krajíček, No counter-example interpretation and interactive computation, in: Logic From Computer Science: Proc. of a Workshop held November 13–17, 1989, Mathematical Sciences Research Institute Publication No. 21 (Springer, Berlin, 1992) 287–293.

[9] J. Krajíček, Fragments of bounded arithmetic and bounded query classes, Trans. AMS 338 (1993) 587–598.

[10] J. Krajíček, P. Pudlák and G. Takeuti, Bounded arithmetic and the polynomial hierarchy, Ann. Pure Appl. Logic 52 (1991) 143–153.

[11] R. Parikh, Existence and feasibility in arithmetic, J. Symbolic Logic 36 (1971) 494–508.

[12] P. Pudlák, Some relations between subsystems of arithmetic and the complexity of computations, in: Logic From Computer Science: Proc. of a Workshop held November 13–17, 1989, Mathematical Sciences Research Institute Publication No. 21 (Springer, Berlin, 1992) 499–519.

[13] G. Takeuti, Sharply bounded arithmetic and the function $a \dot{-} 1$, in: Logic and Computation, Proc. of a Workshop held at Carnegie–Mellon University, 1987, Vol. 106 of Contemporary Mathematics (American Mathematical Society, Providence, RI, 1990) 281–288.
[14] A.J. Wilkie and J.B. Paris, On the scheme of induction for bounded arithmetic formulas, Ann. Pure Appl. Logic 35 (1987) 261–302.
[15] D. Zambella, Notes on polynomially bounded arithmetic, J. Symbolic Logic, to appear.